

N89 - 20697

A SURVEY OF
INTELLIGENT TUTORING SYSTEMS:
IMPLICATIONS FOR
COMPLEX DYNAMIC SYSTEMS

BY

ROSE W. CHU

JANUARY 17, 1989

(DRAFT COPY)

Introduction

The purpose of this paper is to provide an overview of the research in the field of intelligent tutorial systems (ITS). More specifically, the various approaches in the design and implementation of ITS will be examined and discussed in the context of problem solving in an environment of a complex dynamic system (CDS). Although there have been several excellent sources of discussion on the work in ITS (Sleeman and Brown, 1982; Wenger, 1987; Psotka et. al., 1988), the motivation for the paper stems from the need to consolidate the findings in the research to a specific domain of interest. In the Center for Human-Machine Systems Research at the Georgia Institute of Technology, one of our interest and focus of research is the application of ITS to complex dynamic systems.

Several relevant topics will serve as the background to the actual study on the numerous ITS. First, issues pertaining to a CDS will be considered. Next, the nature of human problem solving will be discussed, especially in light of a CDS. Then, an overview of the architecture of an ITS will be provided as the basis for the in depth examination of various systems. Finally, the implications for the design and evaluation of an ITS will be discussed along with some concluding remarks and thoughts.

Complex Dynamic Systems

With the advancement of computer technology, the trend towards more complex systems has posed immediate challenges to the field of human-machine interactions due to the changing role of an operator in his work environment. Rasmussen (1986) has cautioned that automation made possible in these systems do not render the human obsolete, rather, only the previous responsibility of the human operator in low level system controls have now been replaced. In fact, Wickens (1984) points out three objectives of automation. It allows the execution of functions in a system that an operator cannot perform due to inherent human limitations. Also, automation may take over functions that do not involve the best of human capabilities or are within human limitations but are too taxing. Instead of totally taking over, another objective of automation may be to provide assistance to the operator in achieving the above functions .

An operator's new role as a consequence of automation, has generally been discussed under the term supervisory control. According to Sheridan (1976) "the supervisory control paradigm applies to situations where a person allocates his attention among various graphical or alphanumeric displays and intermittently communicates new programs to a computer which itself is in continuous direct control of a physical process." An operator engaged in supervisory control (thus, he is the supervisory controller) must deal with multi-task, multi-goal and multi-person environments (Baron, 1984). The various activities of a supervisory controller have been characterized in different but consistent ways.

Sheridan (1984) considers the planning, teaching, monitoring, intervening and learning modes of a supervisory control task. Wickens (1984) discusses the control versus diagnostic nature of the operator's task. Baron (1984) categorizes the activities into planning, monitoring, situation assessment, decision-making, control and communication. Salvendy (1984) breaks down the task into monitor, control, interpret, plan and diagnose. Yet another simple dichotomy of a supervisory control task is that of monitoring versus troubleshooting. Generally speaking, these activities focus on the cognitive behavior instead of the psychomotor performance of the operator.

These tasks imply requirements at a level not considered before (Rasmussen, 1986). For example, an operator must be trained differently in order to meet the demands of his new tasks. An operator must possess knowledge and understanding about the system at a sufficient depth in order to handle both normal and abnormal situations. Moreover, with automation comes a new set of problems (Wickens, 1984). An operator has to deal with an increased monitoring load in face of a more complex system that now have many additional interacting components. On the other hand, an operator may exhibit too much trust in the automated subsystems, resulting in a false sense of security that in turn affects his job performance. There is also the potential problem of "out-of-the-loop familiarity". This problem arises when an operator is taken out of the normal control-loop replaced by automation, and thus interacting less with the system and becoming less familiar with system states. Consequently, the operator may be less able to handle system trouble. Although automation eliminates some low-level human error, it also introduces other high-level errors associated with an operator's job. Finally, many tasks that previously involve the cooperation of two human operators may now be replaced by a less personal operator-machine team.

How is a CDS distinguishable from other systems? Baron (1984) cited the following features for a system that require supervisory control:

- the system is very high-tech, large scale, expensive and risky in nature
- the system involves many complex and dynamic processes with many controllable outputs
- the system has many subsystems
- many but not all aspects of the system are automated
- manually controllable variables have slow response, in contrast to automatically controlled and rapid changing variables
- the demands on the system is driven by events
- there is a need to communicate among operators and with other system units
- an operator at times have to follow a predetermined set of instructions during some predictable situations.

A lot of work has been done to model the human supervisory controller (Sheridan, 1984; Baron, 1984; Rouse ? etc). In addition, Rasmussen (1986) has recently provided a valuable framework for understanding and designing supervisory control systems. The next

section discusses a framework for studying the human problem solving behavior in a CDS.

Problem Solving Strategies and Models

Human problem solving has been the subject for research in many aspects of human-machine systems. With respect to a CDS, the tasks of a supervisory controller concern that of solving problems in various situations. Much of the research in this area has focussed on the identification of the different strategies that an operator used in problem solving. Salvendy (1984) cited eleven strategies identified in the literature. A brief discussion of each method is given below.

**** NEED TO FIND DEFINITIONS**

Backward search (Simon and Simon, 1978) ...
means-end analysis ...
and hill-climbing...(Newell, Shaw and Simon, 1960),
scan-and-search (Simon and Newell, 1971),...
progressive deepening (DeGroot, 1965) ...
and symptomatic search (Rasmussen, 1981; Wortman, 1971) ...

Application of examples (Anderson, 1981) refers to our ability to solve a new problem by referring to an example of an already solved problem. Solving problem by analogy (Mayer, 1981; Gentner and Gentner, 1983; Rumelhart and Norman, 1981; Carroll et al, 1981) involves using solutions in a familiar domain to solve a problem in the new domain. There are some problems that are solved by mental simulation (Hollan et al, 1980). This means that we envision in our mind a scenario surrounding a fact or a problem which may or may not exist. When the problem solving situation is that of fault diagnosis, Rasmussen (1981) points out that an operator may use a strategy called topographic search. In this situation, the operator has a mental model of the normal functions of the system which is mapped against a problem to determine where a system function may have failed. Finally, Rasmussen (1981) also noted three general types of problem solving behavior: skill-, rule- and knowledge-based performance. Skill-based behavior are sensorimotor type performance that is very automatic. Rule-based behavior follows some prescribed procedure in solving a problem. For complex and/or unfamiliar problems, an operator has a goal in mind and plans his actions to achieve the goal based on his model of the environment surrounding the problem. This is knowledge-base behavior.

In the study on human problem solving in fault diagnosis tasks, several models were proposed (Rouse and Hunt, 1984). These models have both prescriptive and predictive value in an attempt to understand the nature of human problem solving. First, models of complexity suggest that measures of complexity should take into account both the problem and problem solver. Second, the theory of fuzzy sets may be used to model the

decision-making component in a problem which involves more than yes/no answers. Rouse and Hunt also proposed a rule-based model where an operator is modelled to solve a problem based on a set of situation-action heuristics. Next, a fuzzy rule-based model accounts for problem solving with highly context-sensitive rules. Lastly, a overall model considers problem solving to consist of three levels of behavior: recognition and classification of the problem situation, planning towards a solution to the problem, and execution and monitoring of the planned actions.

Complexity in Problem Solving

In the previous section, problem solving was discussed from a prescriptive point of view. The question remains as to what is it that makes problem solving complex? Woods' (1988) approach to the psychology of human behavior in complex problems is especially relevant to our interest in ITS. The reason is that his particular approach provides us with insights to determining the goals of an ITS -- what do we want the ITS to teach an operator in a complex dynamic system. The questions that Woods addressed include: what is complexity? how can we map the inherent complexities of particular worlds? what cognitive demands does a world impose on problem solvers? The rest of this section summarizes Woods' discussions and "answers" to these questions.

Complexity is not an entity by itself, it is a characteristic of a situation. Problem solving situations where complexity becomes an issue can be thought of as interactions between three components. First, there is the world or domain of interest to be acted on because of the problem. Next, there are one or more agents acting on the world in an attempt to solve the problem, in other words, the problem solver(s) and finally, the external representation of the world available and perceived by the agent(s). Problem solving situations become complex if the inherent characteristics of the world impose on the agent(s) cognitive demands that affect the adequate performance in various situations.

From the perspective of the world, Woods defines four dimensions of complexity that contribute to the cognitive demands of that world. First, a world can be characterized by its dynamism; this includes how event-driven is the world and how much do various tasks compete over time. The number of parts and the extent to which these parts interconnect and interact in a domain provide the second dimension of complexity. A world is also characterized by its level of uncertainty in the data that describes the state of the world. Finally, the amount of risk involved in a world is the fourth dimension of complexity. Thus, every domain or system can be analyzed along these dimensions. With respect to the earlier discussion on complex dynamic systems, it is observed that the four dimensions are consistent with the previous characterization of CDS. In general, a CDS is a world that is very dynamic in nature, has many interconnecting and interacting parts, and involves some degree of uncertainty and risk.

So what is the impact of such a world on the cognitive demands and situations that the problem solver(s) will have to face? That is, a world that is defined relatively high on all the four dimensions above? The rest of the discussion will focus on the consequences of each dimension of domain complexity on the problem solving environment confronted by the agent(s).

In a dynamic and event-driven world, problem solving extends over time and solution to a problem may be long term and changing. Moreover, problems are interrelated: the plan(s) of actions to one problem influence the state or solution to other problems. New events or disturbances may occur at any time to affect a problem and/or how it is being solved. Consequently, a problem solver must have the cognitive skills to cope with the above situations. A dynamic world demands that a problem solver must be adaptive in two major ways. First, the problem solver must be able to make predictions about potential possibilities of how the system may behave. Second, the problem solver must be sensitive to the effects of new events or disturbances and be responsive to these effects in terms of his understanding of the world and his plans towards a problem solution. To support these skills, the problem solver must possess knowledge about the world, its different states of behavior and its potential changes between states.

When a domain of interest is characterized by many interacting parts, there are several aspects that contribute to the complexity of the problem solving environment. If a problem solver is faced with a system with a large number of parts, he must learn to manage his time among various tasks that involve different parts. The problem of divided attention is intensified when the domain is also dynamic; the problem solver needs good prospective memory that enables him to come back to a task at a later time. However, if the parts in a system are intricate objects by themselves, it becomes very important for the problem solver to have a good understanding of the workings of these parts. In fact, a complex part is a system in itself and serves as a subsystem to the larger, global system.

When numerous components of the domain are extensively interconnected, several consequences are inevitable. First, actions carried out by the system operator to attain a particular effect may produce undesirable side effects. Similarly, errors and faults can propagate within various parts in the system. Also, such a world is a prime candidate for situations with conflicting and competing goals. In order to perform effectively the reasoning involved in such an environment, the problem solver must have knowledge about how different parts interrelate, affect and constraint each other in achieving different goal states. When faced with a situation with multiple faults, a cognitive demand on a problem solver is that of problem formulation. Essentially, the problem solver must be able make judgements about the problem to focus on based on his assessment of the situation and his knowledge about the system and its components. Another cognitive skill that a problem solver should possess

is disturbance management, particularly when the domain is also dynamic in nature. This skill helps the problem solver deal with the effects of the disturbance(s) at the moment and correct the crisis in the long run. Yet another cognitive demand on the problem solver involves diagnostic situations. The problem solver must have sufficient diagnostic skills to avoid errors such as fixation of a single explanation to account for the state of the world, treatment of interrelated problems as independent and oversimplification of the interconnectedness that exists among the various subsystems of the world.

When the domain is high on the uncertainty dimension of complexity, data available to the operator may be unreliable and that a given datum may be evidence to more than one part or state of the world. As a consequence of the former situation, a problem solver must have sufficient inference abilities to collect and integrate the erroneous data in order to explain a particular state of the world. To cope with the latter situation, the operator must have good reasoning skills to correctly map the evidence from the data to the state(s) these data testify to. Thus, the prerequisites to these skills include the problem solver's adequate knowledge on the various mappings of evidence to state(s). If uncertainty is coupled with dynamism, the task of the operator to collect evidence is compounded by two factors. First, not all data about the state of the environment are accessible at a given time. Second, the operator needs to weight the potential benefit of the information to be acquired with the cost or effort in the acquisition process. As a result, the problem solver needs to know different methods for collecting data; that is, he must know when and where to look for data. (** mention about monitoring aspect of the supervisory controller **) He must respond to and check for system events that unfold over time for evidence of a state of the system. Moreover, he must have adequate knowledge about the states of the system to initiate actions that support evidence gathering. In general, the cognitive demand to cope with large amount of data and information is part of problem formulation, where the problem solver must have the ability to discriminate and attend to relevant data in order to arrive at a solution. Correct utilization of the evidence surrounding an incident will avoid the potential of solving the wrong problem.

Finally, when the world is complicated by the presence of risk, the problem solver, in general, is constantly making decisions that takes into account the cost of a particular choice of action(s) to the overall state of the world. In addition, the problem solver must be concerned with not just expected and common situations, but infrequent situations with damaging results to the system.

In the final analysis, Woods emphasizes the importance of the above approach in the understanding the complexity of a problem solving world. The various demands and situations have strong implications on the other two elements of a problem solving situation, namely, the representation(s) of the world to the problem-solving agent(s) and the cognitive processing

capabilities of the agent(s). The breakdown on the different cognitive demands and situations also provide the basis for understanding the effectiveness and appropriateness of the numerous problem solving strategies that were discussed previously. In accordance to theme of this paper, a global and ideal goal of an ITS designed for a complex dynamic system is to teach an operator all the cognitive skills that he requires to cope with the various cognitive demands and situations that arised due to complexity of the domain. The ITS should also instill into the operator all the knowledge about the system that he will need to support the skills. Questions such as how these skills is taught, and how much of the knowledge should be or can be taught explicitly are yet to be explored and answered.

Architecture of an Intelligent Tutorial System

- * basic elements are domain expertise, student model, pedagogical expertise and interface (Wenger, 1987)

- * similar breakdown by Fath (1987): task model, student model and instructional module. Interface is part of simulation.

*** according to Wenger

**** domain expertise**

- * functions

- has two functions: as a source of knowledge and a standard for evaluating the student's performance
- as a standard, must be able to generate multiple solutions to a problem
- as a source of knowledge, there is a trade off between representing knowledge of expertise as a curriculum (static) versus as a model (dynamic)

- * aspects of communicability

- domain knowledge includes pieces of information that are specifically used for instructional purposes (the learning process)
- issue of transparency of the expert module: how inspectable and interpretable are the reasoning steps to the final results
- issue of psychological plausibility of the expert module: how similar is the expert module's performance as compared to the human's.
- choice of viewpoint of the domain to be taken by the expert module should match that of the student. this is a limitation as compared to human expert's adaptability to various student's viewpoints.

**** student model**

* **information:** how accurate and well covered is the information contained in the student model

- information to interpret a student's behavior
- information to determine the knowledge state of the student based on the interpretation of his action
- explicit representation of the misconceptions a student may have about the domain
- information to explain how these misconceptions may have come about

* **representation:** language of representation must accomodate for incorrect knowledge of the student. language for expertise is thus not sufficient.

- neutral primitives: granular enough to account for both correct and incorrect knowledge in domain. language itself does not carry "correctness".
- error primitives: enumerative approach-- information about errors and misconceptions for a particular domain of students empirically collected and treated as primitives of the language.
- language is such that the student model should be runnable: model can generate predictions about the behavior of a student in a particular context.

* **diagnostic process:** accounting for data to form and update student model;

involves formulation and evaluation of competing hypotheses.

- assignment of credit and blame: intrepertation of actions may be top-down or bottom up. search for the student model may be model-driven or data-driven.
- diagnostic process should be robust to noise from three sources: student model is an approximation of the actual student; students are never perfectly consistent in their actions; learning factor may alter the truth about the knowledge state of a student.
- the diagnostic process may be active during a session by taking over a session and requiring the student to do stuff for diagnostic purpose. or the process may be passive; it observes and analyzes the student's action silently in the background. the process may be a mixed too.
- diagnosis may be interactive in nature if a student is involved in explaining his own behavior (but people are not good at doing that) or may be inferential where a student is excluded totally in the diagnostic process. a mix may be preferred.

**** pedagogical expertise:** knowledge about how to communicate knowledge

* **didactic process**

- represent pedagogical knowledge as rules versus principles
- global decisions affect the sequencing of instructional episodes

- local decisions affect the "when, what and how" of intervention. also includes decisions on guidance in performance, explanations of phenomena and remediation.

*** degree of control**

- monitor student's actions, but system never takes over
- mixed-initiative: control shared by both student and system
- guided-discovery learning or coached activities: student is in full control

**** interface: final form of communication**

*** function**

- interface should have conversational capabilities between the student and the system
- form of communication may involve language processing
- more popular form due to advanced technology is the use of computer graphics in representation

*** desiderata (what is desired in the interface)**

- should be clear and understandable in presenting system's topic
- should be explicit about system's capabilities
- should be easy and attractive to use for the student

***** these breakdown does not necessarily correspond to distinct modules in an ITS. also decisions about any of these issues in any one component will very likely affect those made for other components

Models of Intelligent Tutoring Systems

The outline for each discussion of a model is organized as follow:

A. Description

Any interesting or important general facts about the model is mentioned here. The methodologies or approaches used for each of the component of the ITS are identified under the following subheadings:

- domain expertise
- student model
- pedagogical expertise
- interface

B. Implications for Complex Dynamic Systems

What is applicable and what is not and why with respect to the dimensions of complexity will be addressed in this section.

C. An example in the GT-MSOCC Domain

The issues raised above will be illustrated and discussed in the context of an existing complex dynamic system called GT-MSOCC.

(1) SCHOLAR (Carbonell, 1970)

A. Description

SCHOLAR is considered the first intelligent tutoring system ever developed. Carbonell pioneered the artificial intelligence approach to ITS where knowledge is explicitly encoded. This approach replaced the traditional frame-oriented paradigm.

Domain Expertise

The system applies to the geography of South America. This domain knowledge is represented in a semantic network. The nodes on the network represent relevant objects and concepts that the system knows about. These objects are linked together hierarchically in the network.

Student Model

A early version of the "overlay" model (discussed in more details later) is used. The network can be used to represent the knowledge of an ideal student. Evaluations on a student's actual performance are identified with the concepts in the network that are taught.

Pedagogical Expertise

SCHOLAR does not have any sophisticated tutorial strategies. Its main concern in this respect is to select relevant topics for discussion based on the distance between nodes on the network and the notion of relevance tags of these nodes. Decisions are thus very local and at times random. The student and the system interact in a mixed-initiative dialogue mode.

Interface

The form of communication is textual. A template matching process is use to generate and parse simple sentences.

B. Implications for CDS

For factual knowledge such as geography, the notion of nodes and links can be readily defined. However, for an operator in a complex dynamic system, facts alone are not sufficient; he needs to possess procedural knowledge to carry out his tasks as a supervisory controller. Exactly what the nodes and links mean is not so clear for "how to" type information.

Another important aspect of a complex dynamic system that cannot be represented with a semantic network is dynamism. Specifically, such a network cannot accommodate the passage of time to reflect the potential changing states and behavior of a system. Such knowledge is crucial for an operator in developing his adaptive skills (recall Woods' discussion).

It is conceivable that semantic nets can be used to represent one "viewpoint" of a CDS in an ITS. For example, the complexity of the system in terms of the number of parts and their interconnectedness could be represented by several semantic networks at various levels of abstraction.

C. An Example in GT-MSOCC

One of the operator's function is to manually configure a mission upon request. In order to correctly carry out such a function, an operator must be taught to follow a sequence of plans. Such procedural knowledge would not be adequately represented in a semantic net.

However, part of the training of the operator is to acquire some background knowledge about the system. Factual knowledge such as the various mission configurations, the list of equipments needed by each mission and the maximum number of missions supported at any time could be represented as one or more semantic networks. The goal of the ITS at this point would be to make sure that the operator knows these facts about the system before moving on to the various operator functions. Somehow, the representational scheme used beyond this stage should be connected to the semantic network(s) for smooth transition and consistency.

(2) WHY (Stevens and Collins, 1977)

A. Description

Domain Expertise

WHY represents its domain knowledge in rainfall processes with hierarchical scripts. The authors attempt to capture both temporal and causal relations between typical sequences of events in these meteorological processes.

Student Model

There is no student model. A student's performance is evaluated independently.

Pedagogical Expertise

The tutorial strategy implemented in WHY is the Socratic method. In this method, a tutor asks the student questions to guide him in developing skills and principles for managing hypotheses and drawing relevant inferences from data collect. The strategy is captured in a set of tutorial rules that deals with local decisions about the appropriate questions to ask based on the student's last response. No global tutorial goals are considered in these decisions.

Interface

The dialogues between the tutor and the student is strictly textual. The natural language is processed in a similar fashion as in SCHOLAR.

B. Implications for Complex Dynamic Systems

The issues that evolved from the two major weakness of WHY have been discussed in length by Wenger. These issues will be explored further with respect to complex dynamic systems.

Considering global tutorial goals

In the rainfall domain, Stevens and Collins (1977, 1982) examine the higher-order goals of a human tutor that influence his tutorial decisions. They suggest that such goals must be incorporated into the pedagogical module of an ITS. To consider such goals is then to identify the teaching goals in terms of what a student is supposed to learn. The choice of a pedagogical approach should be consistent with these goals. It is possible and likely, especially with respect to complex dynamic systems, that the approach selected will embody more than one tutorial strategies to achieve all the pedagogical objectives.

In terms of the the kind of cognitive situations an operator will encounter and the type of skills needed to cope with these situations, when and how may the Socratic method be applicable? One possible direction is to

isolate a particular cognitive situation and tutor the operator/student to develop the corresponding skills in a Socratic style. The situation, which is a "case" in Socratic terms, could be presented to the student in a scenario of system events. The tutor proceeds to ask meaningful questions based on the student's actions or responses.

There are several problems that immediately come to mind. In a complex and dynamic world, the various cognitive situations overlap and interact with each other among all dimensions of complexity. Thus, there is no assurance that the skills acquired from two isolated situations will translate to the skills required to manage a single incidence with cognitive demands of both situations. Because the world is dynamic, events are evolving in "real time". As a result, a tutorial dialogue occurring within a scenario must avoid being too obtrusive to the extent of becoming unnatural. Another potential problem is that important events in the scenario may be missed while the dialogue is in progress. Intuitively speaking, it is not feasible to use only the Socratic style of teaching when the domain of interest involves a complex dynamic system. It seems that there may be skills more appropriate than others, and that there may be a more suitable time in the student's learning process than others to apply the Socratic method.

Represent domain knowledge from multiple perspective

The fact that scripts reflect only linear relations between events is even more profound a limitation in complex dynamic systems. Large number of components interact with each other in nonlinear and often unpredictable ways. In order for a student to develop skills to handle problems such as divided attention and prospective memory, the representation scheme chosen for the ITS must account for such nonlinearities.

Another limitation of script-based representation is that only global aspects of a process are captured in temporal and causal terms. The suggested functional perspective of the domain knowledge is particularly relevant in a complex dynamic system. The operator needs to have knowledge about the workings of each component and how it affects and constrains other components in the system. This knowledge supports the operator's many skills such as problem formulation in situations with multiple faults and conflicting goals. That is, both the "x causes y when" aspect and the "how x causes y and why" aspect of the domain knowledge must be captured in the expert model of an ITS.

Besides the above limitations, scripts are not suitable for expressing complex dynamic worlds for reasons characteristic of such worlds. Scripts are good for stereotypical sequences of events. In a complex dynamic system, from the perspective of a supervisory controller, the cause for concern is more for non-stereotypical sequences of events instead. Operators must know not just what normally happens to the system over time, but also what to do in novel situations. Skills in disturbance management and reasoning and inferencing abilities are required of these operators. In any case, the dynamic nature of such a system makes the task of defining all possible sequences of events a very exhaustive and impractical ordeal. Moreover, the uncertainty dimension (in terms of

system behaviors) makes the prediction of all potential sequences of events unrealistic. With regards to the issue of psychological plausibility, it is certainly true that experts do not have a script for every possible situation in order to solve different problems.

To the extent that the idea of multiple viewpoints in the representation of domain knowledge is believable, the form of communication of these viewpoints must go beyond just textual interface. The advance in computer technology make the use of visual and graphical techniques in interface design a very viable option (more on this is discussed in later models).

C. An Example in GT-MSOCC

Consider the possibility of implementing a Socratic style tutor for GT-MSOCC. A session (or a scenario) in GT-MSOCC has the goal of teaching the operator how to troubleshoot endpoints for software failures. The operator's actions and responses are evaluated such that the tutor can pose appropriate questions. The following is a sample list of what might happen:

1. The operator types "display msocc sched". Then there is a long pause...
2. The tutor decides to ask a question: "Do you think you need to check endpoints now?"
3. If the operator answers "yes", the tutor predicts the operator will next execute commands that support the goal to check endpoints (eg. display vip telem).
- 3a. The tutor then asks "Why do you need to see tac telem page?" to explore the operator's understanding of the task.
- 3b. The operator may then answer "Because vip3 is an endpoint equipment for the mission ERBE".
- .
- .
- .
4. If the operator answers "no" to question in item 2, the tutor may ask "why not?"
- 4a. student may answer "because"

.

.

.

(3) METEOROLOGY Tutor (Brown et al., 1973)

A. Description

This project launched the work on qualitative models and set the grounds for subsequent research on SOPHIE (next section).

Domain Expertise

As the name of the system implies, the domain of application is meteorology. The core technique represents the causal knowledge about meteorological processes in a qualitative simulation model. Sequences of events in each process are simulated via a finite-state automata. The semantic network approach used in SCHOLAR is also implemented in this system to represent meteorological concepts.

Student Model

No effort is directed to modeling the student here.

Pedagogical Expertise

The tutor is a question-answering system. Questions about factual knowledge from the student are answered in a similar way as in SCHOLAR. To generate explanations for a question about a process, an inference tree is built dynamically from the simulation model. This inference tree describes the temporal and causal relations between events as related to the question.

Interface

The tutorial dialogues between the tutor and the student is carried out in natural language form. A simple process of keyword matching is used to extract the context of a question. Answers to questions about processes are constructed by joining successively predefined text units that reside in each state of an automata.

B. Implications for Complex Dynamic Systems

Operator Control Model (Miller, ?) and Operator Function Model (Mitchell, 1987) are two modeling frameworks that involve networks of finite-state automata. The task of predefining all possible series of events is replaced by the identification of system states. The dynamism of such systems can then be captured in the state transitions within the network. Thus, the idea of a dynamic process model is especially befitting with regards to complex dynamic systems.

The idea of dynamic generation of explanations may be used to consider a question-answering option for an ITS. The student selects this mode to acquire or review knowledge about the system. Such an option can only be supplementary to the actual teaching that is needed to assist the student in developing the appropriate skills in terms of a complex system.

- need better nlp instead of prestored text. in fact, should be able to take advantage of visual methods in presenting the answer (eg. showing the inference tree where answer is).

- idea of multiple representations supports the idea of multiple viewpoints.

- domain representation affects pedagogical decision and vice versa. that is, teaching goals also affect how we want knowledge to be expressed. what viewpoints or mental models do we want student to develop of the system? physical, functional, causal?

C. An Example in GT-MSOCC

- OFM methodology represents operator functions in a heterarchical/hierarchical network where state transitions reflect system triggering events. A tutor for GT-MSOCC may use the OFM for pedagogical decisions in exploring the student's understanding of the system and his task. Illustrates the dependency between domain representation and pedagogical strategies.

- since we already have OFM, may include a q/a mode operator can choose. Operator may ask questions relating to a system request or message, its effects, and/or how to fix the problem. Answers may be explanations, or even suggested steps or actions. Not really a tutor implemented. do not know if student is actually learning.

- the use of the blackboard for implementing OFM is one way to make model explicit. thus, operator can view the blackboard and see what he is expected to do.

(4) SOPHIE (Brown et al., 1974, 1976, 1982)

A. Description

Domain Expertise

The domain of application for the entire SOPHIE project is the troubleshooting of electronic circuits. Troubleshooting skills involve the ability to collect various measurements, to hypothesize the potential problem areas and to test such hypotheses.

SOPHIE-I and SOPHIE-II represent the domain knowledge in multiple ways. A simulation model represents the mathematical model of the circuit. Procedural knowledge is captured in a set of specialists based on this model, while declarative knowledge is reflected in a semantic network.

In SOPHIE-III, domain knowledge is represented in two separate module: the troubleshooting expertise and the electronics expertise. The troubleshooting expertise has general troubleshooting knowledge for managing a set of hypotheses. The electronics expertise has both general electronic knowledge and circuit-specific knowledge represented in three different levels: components model, production rules and behavior trees each linked with a different reasoning mechanism and input information.

Student Model

Although considered, this portion of the research was never implemented.

Pedagogical Expertise

The pedagogical paradigm of the SOPHIE project is to provide a reactive learning environment for the student. In such an environment, the student has the opportunity to test his ideas and knowledge, and receive constructive feedback and advice.

In SOPHIE-I, pedagogy consists of generating meaningful feedback to a student's action by making inferences based on the knowledge embedded in the simulation model. An articulate expert troubleshooter in SOPHIE-II explains the reasoning and strategies underlying these inferences. The representational scheme in SOPHIE-III works as an inference engine to reflect human-like reasoning. The idea is to use this engine for coaching and modeling the student in an active environment. Unfortunately, this part of SOPHIE-III was never completed.

Interface

SOPHIE and the student interacts via a very robust natural language interface. The natural language processing is implemented with semantic grammars. The idea is to represent a sentence based on domain-dependent semantic categories instead of its syntax.

B. Implications for Complex Dynamic Systems C. An Example in GT-MSOCC

(5) STEAMER (William, Hollan, Stevens, 1982)

A. Description

This project pioneered the notion of graphical simulations in training systems. Projects such as the Intelligent Maintenance Training System (Munro et al., 1985) and the Recovery Boiler Tutor (Woolf et al., 1986) have been influenced by STEAMER.

Domain Expertise

The domain of application is operating steam propulsion plants in large ships. The model of the domain knowledge is purely mathematical. From the knowledge communication perspective, STEAMER does not really have a model of the expertise.

Student Model

STEAMER does not have a student model (?)

Pedagogical Expertise

STEAMER presents the steam propulsion plant in an interactive and inspectable graphical simulation form. The student can manipulate various aspect of the simulated plant and examine the effects of his actions. The pedagogical goal is to provide a means for the student to acquire a mental model of a complex physical system and at the same time learn to operate such a system.

To further support this goal, two other modules are implemented. When a student is running a particular procedure, the tutorial module can furnish feedback in the form of explanations based on the graphical abstractions that define the simulated plant. Another module called the feedback minilab allows the student to experiment with different control devices. The student can put together the components for a device and STEAMER will test it by integrating the simulated device with the rest of the system.

Interface

Within the STEAMER's graphical interface, the system and the student interact through simple text processing. (eg. menus and options).

More importantly, the graphical description of STEAMER's simulation model initiated the principle of conceptual fidelity. The goal is to present a conceptual view and not the physical view of a complex system. This view when presented to the student is considered faithful to the actual system if it expresses the same view possessed by experts. Such a view should reflect the mental model that experts use when they reason about the system.

- B. Implications for Complex Dynamic Systems**
- C. An Example in GT-MSOCC**